Risk-Controlled Selective Prediction for Regression DNN models*

*Note: Sub-titles are not captured in Xplore and should not be used

1st Wenming Jiang
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Ying Zhao
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3rd Zehan Wang
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—Deep Neural Network (DNN) regression models have been successfully utilized in numerous fields. However, selective techniques, also known as reject option, has yet been mainly considered in classification NNs, comparing to the limited work in regression NNs. In this paper, we consider the selective regression problem from a risk-coverage view, and propose a method to construct a selective regression model given a trained DNN model under a desired regression error risk. Then, we propose to utilize blending variance to quantify uncertainty in regression NNs. We evaluated both the proposed uncertainty functions and the selective regression method for two real-world applications, the tropical cyclone intensity estimation problem and the apparent age estimation problem. Our proposed methods achieved promising results. For example, for the TC intensity estimation problem, our selective regression model achieved a RMSE value of 9.5 for 75% test coverage with a guided confidence level of 0.05, whereas the overall RMSE achieved by the state-of-the-art model is 10.5.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Deep neural networks (DNNs) have been widely used for various regression problems, such as tropical cyclone (TC) intensity estimation [1]–[3], age estimation [4], [5], wind power prediction [6], and pain intensity estimation [7] and so on. Applying such models in real world applications often requires a control on regression errors on individual samples. For example, TC intensity estimation uses satellite images of tropical cyclones to estimate their intensities. When using these models for weather forecasting, a large regression error on a single satellite image may underestimate the rank of a tropical cyclone and cause significant casualties and economic losses. Hence, measuring model uncertainty is important and we do not want to accept all prediction results without doubt. Instead, we would like to reject the predicted results with high uncertainty (and consult them with domain experts), which is called predicting with a reject option [8].

Most of the existing works on selective prediction focused on the selective classification problem and various uncertainty functions for classifiers to construct selective models [9]–[12]. Given a classification model and a function measuring model uncertainty, selective classification models trade-off between misclassification rates and rejection rates to achieve higher classification accuracy on as many as possible input samples. In particular, [11] and [12] put forward a risk-coverage framework, under which selective classification models are constructed to maximize the selective coverage with a guaranteed risk bound.

However, the existing selective prediction methods for the classification problem cannot be used directly to solve regression problems. We should first acknowledge that a regression problem can be transferred to a classification problem and solved by DNN models. For example, in the TC intensity estimation problem, we can divide the TC intensity into a number of categories (*e.g.*, 18 TC ranks) and use classification DNN models to solve it [13]. However, the evaluation of classification performance is different from that of regression performance. In the classification problem, we commonly use classification accuracy, such as misclassification rate, to evaluate classification results, whereas in the regression task, we usually employ regression loss functions to measure the magnitude of errors, such as Mean Square Error (MSE) or Mean Absolute Error (MAE). Since the existing selective classification methods and theoretical bounds are derived for optimizing classification accuracy, they cannot be used directly for optimizing regression loss functions.

In this paper, we focus on the selective prediction problem for regression DNNs, and made the following contributions:

- For a given regression model f, a confidence level δ, and a desired regression error target r^*, we propose a method to construct a selective function g, such that the selective regression model (f, g) can achieve maximum coverage while keeping expected regression error no larger than r^* with probability $1 - \delta$.
- We purpose a new uncertainty function for rejection, *Blend-Var*, which measures the variance of multiple predictions of a single input sample (such as an image) when blending with rotation, reflection, shift and so on.

Identify applicable funding agency here. If none, delete this.

- We evaluated our selective regression models with the proposed uncertainty functions on two real-world applications and achieved promising results. For example, for the TC intensity estimation problem, our selective regression model achieved a RMSE value of 9.5 for 75% test coverage with a guided confidence level of 0.05, whereas the overall RMSE achieved by the state-of-the-art model is 10.5.

The rest of the paper is organized as follows. Section 2 reviews related works. Section 3 introduces the background of the general selective model and Hoeffding's Inequality and its applications. Section 4 proposes the risk-controlled selective regression model. Section 5 presents the experimental results on two real-world applications. Section 6 concludes the paper.

II. Related Work

Selective prediction, or prediction with a rejection option, has been studied for more than a half of century. Early works tackled the problem based on statistical decision theory to trade-off between error and rejection in the recognition problem [8], [14]. Later on, selective prediction models were proposed for various hypothesis classes and learning algorithms, among which selective models for neural networks (NNs) [9], [10] and deep neural networks (DNNs) [11], [12] drew people's attention lately. Most of the existing works focused on the selective classification problem and various uncertainty functions for classifiers to construct selective models. Recently, a deep neural architecture called SelectNet [15] with an integrated reject option was put forward, which is trained to optimize both classification (or regression) and rejection simultaneously.

Among the existing selective NN classification works, there are basically two types of selection models: from a cost-based view and from a risk-coverage view. [9] and [10] tackled the problem from a cost-based view, which defines a selection cost function including both rejection rate and misclassification rate and searches for the selective classification model that optimizes the cost function. The risk-coverage view, applied in [12] and [11], also aims to trade-off between the selective risk and coverage. However, in this framework, selective classification models are constructed to maximize the selective coverage, under the control of a selective risk target.

The uncertainty functions of classifiers are used to reject badly predicted samples. In [9] and [10], a reject threshold was set on the maximal neuronal response in the softmax layer. This mechanism is known as *softmax response* (SR), which would reject a sample, if none of the neuron has a high response value, which means the probabilities of being classified as each class are relatively the same. In [12] and [16], *Monte Carlo dropout* (MC-dropout) was used to estimate the predictive uncertainty in neural networks with dropout [17]. Dropout could be interpreted as an ensemble technique, approximately combining exponentially different networks with shared weights. [16] showed that a neural network with dropout applied before every weight layer is mathematically equivalent to the probabilistic deep Gaussian process approximately. The predictive uncertainty can then be seen as the sample variance of T times stochastic forward passes through the network.

In recent years, deep neural networks have been widely used for the regression problem as well, such as tropical cyclone (TC) intensity estimation [1], age estimation [4], wind power prediction [6], remaining lifetime estimation [18], and pain intensity estimation [7]. However, the selective prediction problem for regression neural networks was just discussed by [15] lately and has not been well studied yet. Different from [15], we tackle this problem from the risk-coverage view using the given DNNs and some uncertainty functions.

III. Background

Let $\mathcal{X}$ be some feature space (e.g., raw image data or d-dimensional vectors in $\mathbb{R}^d$) and $\mathcal{Y}$, the output space. We have a prediction function f, $f : \mathcal{X} \to \mathcal{Y}$. Although in the literature, uncertainty driven selective models were mainly studied and derived for the classification problem, here we introduce the selective model from a more general point of view, *i.e.*, let f be either a classification function or a regression function, and $\mathcal{Y}$ be either a set of categorical labels or a real-valued set.

A. General Selective Model

A selective model is a pair (f, g) [11], where f is the given function, and $g : \mathcal{X} \to \{0, 1\}$ is a *selection function*, which serves as a binary qualifier as follows. For a given sample x, its output is:

$$(f, g)(x) \triangleq \begin{cases} f(x) & if\ g(x) = 1 \\ reject & if\ g(x) = 0 \end{cases} \tag{1}$$

Note that $(f, g)(x) = f(x)$ if $g(x) \equiv 1$, *i.e.*, no sample is rejected and the selective model is the function f itself. We usually utilize an uncertainty function $\kappa_f : \mathcal{X} \to \mathbb{R}$ for f, to measure how well a prediction fits the corresponding ground truth [19]. Using an uncertainty function κ_f and a threshold θ, we can form a selection function $g_\theta(x)$ as follow,

$$g_\theta(x) = g_\theta(x|\kappa_f) \triangleq \begin{cases} 1 & if\ \kappa_f(x) \le \theta \\ 0 & if\ \kappa_f(x) > \theta \end{cases} \tag{2}$$

The idea of selective model is to reject some badly predicted samples so that f can achieve better performance on the remaining ones. The performance of selective model can be considered from a *risk-coverage* view [11]. Formally, let $P(X, Y)$ be a distribution over $\mathcal{X} \times \mathcal{Y}$. The selective coverage of (f, g), defined to be $\Phi(f, g) \triangleq E_P[g(x)]$, is the no-reject-region-rate in $\mathcal{X}$. The selective risk of (f, g) is defined as

$$R(f, g) \triangleq \frac{E_P[\ell(f(x), y)g(x)]}{\Phi(f, g)}, \tag{3}$$

where ℓ is a loss function measuring the loss between $f(x)$ and the true output y of x, $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.

It is hard to obtain $\Phi(f, g)$ and $R(f, g)$ from the unknown distribution. Instead, we can construct a validation set consisting m labeled examples $S_m = \{(x_i, y_i)\}_{i=1}^m$, assumed to be

sampled i.i.d. from $P(X,Y)$, and estimate $\Phi(f,g)$ and $R(f,g)$ on S_m.

Now, given a confidence parameter $\delta > 0$ and a desired risk target $r^* > 0$, the goal is to find a selection function g that maximizes $\Phi(f,g)$ while its selective risk satisfies

$$Pr\{R(f,g) > r^*\} < \delta. \tag{4}$$

B. Measuring Model Risk with Hoeffding Bounds

Hoeffding's inequality [20], Chernoff bound, and Azuma's inequality [21] are the main analytic tools to bound the probability of a large discrepancy between sample and population means. In machine learning, Hoeffding's inequality is often used to ensure the generalization of a prediction function f by bounding the probability of the gap between the expected risk over the distribution $P(X,Y)$ and the empirical risk on a validation set of f, which can be stated in the following lemma.

Lemma 1: Given a prediction function f, a distribution $P(X,Y)$, and a loss function l, assume that $b = max_{P(X,Y)}(\ell(f(x)-y))$ and $a = min_{P(X,Y)}(\ell(f(x)-y))$ are finite and real-valued. If we sample n data i.i.d. from $P(X,Y)$ (*i.e.,* $(x_i,y_i) \sim P(X,Y)$, for each $1 \le i \le n$), then for $t \ge 0$,

$$Pr\{R(f) - R_n(f) \ge t\} \le e^{\frac{-2nt^2}{(b-a)^2}}, \tag{5}$$

where $R(f) = E_P(\ell(f(x),y))$ is the expected model risk and $R_n(f) = \frac{1}{n}\sum_{i=1}^{n}\ell(f(x_i),y_i)$ is the empirical risk over n samples.

Hoeffding's inequality provides loose bounds for estimating model risks. In practice, $b-a$ are often approximated using the validation set, for example, by maximum absolute error minus minimum absolute error of f [22] or adding a few standard deviations to the average error of f [23] to avoid the affection of prediction outliers of f.

IV. Method

We formulate the selective regression problem following the general selective model framework and propose a learning algorithm to obtain selective regression models that are likely to produce better regression performances for a large portion of samples from the input space $\mathcal{X}$.

A. Problem Setting

For a regression model f (such as regression neural network models), the output space $\mathcal{Y}$ is assumed to be real-valued, $\mathbb{R}$. The expected risk of f w.r.t. the distribution over $\mathcal{X} \times \mathcal{Y}$ (*i.e.*, $P(X,Y)$) is $E_P(\ell(f(x),y))$, where the loss function $\ell(f(x),y)$ is usually square error $\ell(f(x),y) = (f(x)-y)^2$ or absolute error $\ell(f(x),y) = |f(x)-y|$. Thus, $E_P(\ell(f(x),y))$ measures the Mean Square Error (MSE) or Mean Absolute Error (MAE) of the regression model f, assumed to be real-valued and finite.

A *selective regression model* is a pair (f,g), where f is a regression model and g is a selection function as defined in Equation 1. Similarly, we form a validation set consisting m labeled examples $S_m = \{(x_i,y_i)\}_{i=1}^m$, assumed to be sampled i.i.d. from $P(X,Y)$. We formulate the selective regression problem as follows.

Definition 1: **Selective Regression Problem**. Given a feature space $\mathcal{X}$, a real-valued output space $\mathcal{Y}$, a distribution over $\mathcal{X} \times \mathcal{Y}$, $P(X,Y)$ (shorted as P), a regression model $f : \mathcal{X} \to \mathcal{Y}$, a validation dataset S_m, a confidence parameter $\delta > 0$, and a desired risk target $r^* > 0$, the **selective regression problem** is to find a selective regression model (f,g) that maximizes $\Phi(f,g)$ while its expected risk satisfies

$$Pr\{R(f,g) > r^*\} < \delta, \tag{6}$$

where $\Phi(f,g) = E_P[g(x)]$ is the coverage of (f,g), $R(f,g) = \frac{E_P[\ell(f(x),y)g(x)]}{\Phi(f,g)}$ is the expected risk of (f,g) and evaluated using regression loss functions, such as MSE or MAE.

Note that the nature of the MSE/MAE-based selection risk reflects the essential difference between regression and classification, which also makes the problem unsuitable for the method proposed in [12].

Although f can be any kind of regression models, in this paper we focus on deep neural networks (DNNs), where existing techniques (such as softmax , dropout [17], and ensemble methods [1]) provide promising ways of measuring uncertainty of f.

B. Selective Regression with Controlled Risk

Given a selective regression model (f,g), let $P_g(X,Y)$ be the projection of $P(X,Y)$ over g, *i.e.*, $P_g(X,Y) \triangleq P(X,Y|g(X)=1)$. The expected risk of (f,g) can be written as

$$R(f,g) = \frac{E_P[\ell(f(x),y)g(x)]}{E_P[g(x)]} = E_{P_g}[\ell(f(x),y)].$$

We can use Hoeffding's inequality [20] to establish the risk bound of a selective regression model (f,g) using a validation set S_m sampled i.i.d. from $P(X,Y)$.

Lemma 2: Given a selective regression model (f,g), a projection distribution $P_g(X,Y)$, and a loss function l, assume that $b = max_{P_g(X,Y)}(\ell(f(x)-y))$ and $a = min_{P_g(X,Y)}(\ell(f(x)-y))$ are finite and real-valued. If we sample n data i.i.d. from $P_g(X,Y)$ (*i.e.,* $(x_i,y_i) \sim P_g(X,Y)$, for each $1 \le i \le n$), then for $t \ge 0$,

$$Pr\{E_{P_g}(\ell(f(x),y)) \ge \frac{1}{n}\sum_{i=1}^{n}\ell(f(x_i),y_i) + t\} \le e^{\frac{-2nt^2}{(b-a)^2}}. \tag{7}$$

The assumption that $b = max_{P_g(X,Y)}(\ell(f(x)-y))$ and $a = min_{P_g(X,Y)}(\ell(f(x)-y))$ are finite is true for many real world applications, such as tropical cyclone intensity estimation and human age estimation, where the output space $\mathcal{Y}$ has natural bounds and $f(x)$ can follow the same bounds.

Suppose an uncertainty function κ_f is available for constructing selective regression models using Equation 2. For a certain selection function g_θ and a validation set S_m, we can filter S_m using g_θ, *i.e.*, keep only samples with $\kappa_f(x) \le \theta$. Note that sampling from $P_{g_\theta}(X,Y)$ is equivalent to filtering S_m using g_θ, since S_m was drawn i.i.d. from $P(X,Y)$. Hence,

we can estimate the expected risk of (f, g_θ), $E_{P_{g_\theta}}[\ell(f(x), y)]$, and the following theorem can ensure this estimation with a guaranteed bound.

Theorem 1: Given a selective regression model (f, g_θ), a projection distribution $P_{g_\theta}(X, Y)$, a loss function l, a validation set S_m, assume that $b = max_{P_{g_\theta}(X,Y)}(\ell(f(x) - y))$ and $a = min_{P_{g_\theta}(X,Y)}(\ell(f(x) - y))$ are finite and real-valued. Let S_θ be the filtered sample set of S_m using g_θ and z is the size of S_θ. Then, for $t \geq 0$,

$$Pr\{E_{P_{g_\theta}}(\ell(f(x), y)) \geq \frac{1}{n}\sum_{x \in S_\theta} \ell(f(x), y) + t\} \leq e^{\frac{-2zt^2}{(b-a)^2}}.$$

Now, we need to search for a uncertainty threshold θ for $\kappa_f(x)$, such that the selection function g_θ maximizes the coverage while its expected risk satisfies $P(R(f, g) > r^*) < \delta$, where δ is the confidence level parameter and r^* is the desired risk. Theorem 1 suggests that the expected risk $E_{P_{g_\theta}}(\ell(f(x), y))$ can be estimated by the empirical risk $\frac{1}{n}\sum_{x \in S_\theta} \ell(f(x), y)$ plus a gap t. Given a confidence level δ, we can obtain an analytic solution of t from the right-hand side of Equation 7 by setting

$$e^{\frac{-2nt^2}{(b-a)^2}} = \delta,$$

which gives us $t = \sqrt{-\frac{(b-a)^2 \ln \delta}{2n}}$.

If we have an ideal uncertainty function κ_f, *i.e.*, for $(x_1, y_1) \sim P(X, Y)$ and $(x_2, y_2) \sim P(X, Y)$, $\kappa_f(x_1) \leq \kappa_f(x_2)$ if and only if $\ell(f(x_1), y_1) \leq \ell(f(x_2), y_2)$, sorting all samples in S_m w.r.t. κ_f also results in a monotonic increasing sequence of $\ell(f(x), y)$. Hence, searching along this sequence can find the maximum number of samples (thus the corresponding θ to make the split) whose empirical risk is less than the desired risk target.

Noticing that $b - a$ is bounded by the difference between the maximum regression error and minimum regression error of f over the data distribution, so t decreases rapidly as n increases and becomes stable after n reaches a number (we call it m_0, *minimum selected sample requirement*), such that the change in n would not bring big change in t and t is small enough compared with both the expected risk and empirical risk.

When $n \geq m_0$, the sequence sorted by κ_f is also a monotonic increasing sequence of $\ell(f(x), y)$ plus t, and we can find the maximum θ whose expected risk is less than the desired risk target. We speed up this process by a binary search strategy shown in Algorithm 1.

In line 2, z_{min} is the starting index to begin the search. We set z_{min} be m_0, the minimum number of samples needed. Lines 4 and 5 define a selection function g_θ, and the first z samples of S_m form the set S_θ for g_θ. With $t = \sqrt{-\frac{(b-a)^2 \ln \delta}{2z}}$ and $\hat{r}_z = \frac{1}{z}\sum_{i=1}^{z} \ell(f(x_i), y_i)$, by Theorem 1 we have

$$Pr\{R(f, g_\theta) > (\hat{r}_z + t)\} \leq \delta.$$

Furthermore, we require $\hat{r}_z + t \leq r^*$ to lead the binary search shown in Lines 9 to 12. When the search terminates and a solution indeed exists, the algorithm finds the maximum θ such that $\hat{r}_z + t \leq r^*$, which also guarantees

$$Pr\{R(f, g_\theta) > r^*\} \leq \delta.$$

Algorithm 1 Selection with Guided Regression Loss (SGRL)

Require: $f, \kappa_f, \delta, r^*, S_m$
Ensure: g_θ, $\hat{r}_z + t$
1: Sort S_m according to $\kappa_f(x), x \in S_m$;
2: $z_{min} = m_0$; $z_{max} = m$;
3: **while** $z_{min} \leq z_{max}$ **do**
4: $z = \lfloor (z_{max} + z_{min})/2 \rfloor$;
5: $g_\theta = g_{\kappa_f(x_z)}$; $S_\theta = \{(x_i, y_i)\}_{i=1}^{z}$;
6: $b - a = Apprx(S_\theta, f)$;
7: $t = \sqrt{\frac{-ln(\delta)(b-a)^2}{2z}}$;
8: $\hat{r}_z = \frac{1}{z}\sum_{i=1}^{z} \ell(f(x_i), y_i)$;
9: **if** $\hat{r}_z + t \leq r^*$ **then**
10: $z_{min} = z + 1$;
11: **else**
12: $z_{max} = z - 1$;
13: **end if**
14: **end while**
15: **if** $z_{max} \geq m_0$ **then**
16: $z = z_{max}$;
17: Calculate new g_θ and $\hat{r}_z + t$, then Output;
18: **end if**
19: Return;

Note that not all input values of δ and r^* lead to a feasible solution. However, when such feasible solution does exist, the algorithm SGRL guarantees to find the selection function g_θ that maximizes the coverage and satisfies the selective risk requirement. In practice, an ideal uncertainty function κ_f may not exist. However, with a proper choice of uncertainty functions, we still can obtain desirable expected risk bounds as shown in our experiments. We also approximated $b - a$ by adding two standard deviations to the average error $(f(x) - y)$ of f on S_θ (as shown in Line 6 of Algorithm 1) as suggested in [23] to avoid the affection of prediction outliers of f. Although in this case, we cannot use Theorem 1 to guarantee the risk bound strictly, it still can be used as a guideline for risk control purposes. We call δ a *guided* confidence level, which was also verified in our experiments.

C. Uncertainty Functions

Consider a regression function f, assumed to be trained for some unknown distribution $P(X, Y)$. In this section we consider two uncertainty functions, one based on the previous work MC-dropout [16], the other one *Blend-Var* based on the variance of blending results, which is first put forward in this paper.

1) MC-dropout: The predictive uncertainty of a neural network with dropout layers can be seen as the sample variance of T times stochastic forward passes through the NN [16]. For a given instance x_0, get T different predictions with dropout

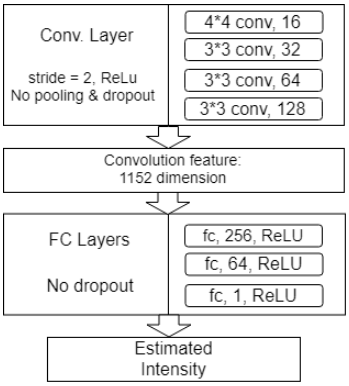


Fig. 1. Network Architecture for TC Intensity Estimation

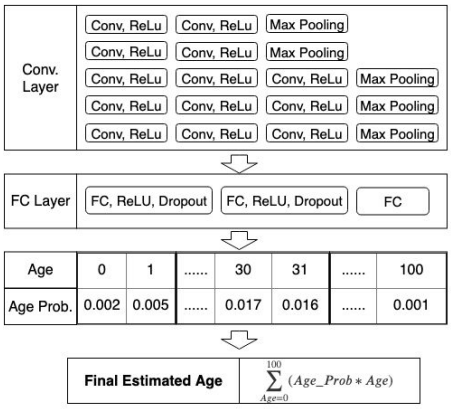


Fig. 2. Network Architecture for Apparent Age Estimation

layers opening. The variance of predictions is used as the uncertainty function κ_f, i.e., $\kappa_f(x_0) = var(f_i(x_0))$, while $i = 1, 2, \ldots, T$ and f_i denote the ith NN differed by the dropout layers.

MC-dropout technique could be used in both classification and regression neural networks. It does not work for those neural networks without dropout layers.

2) Blend-Var: Blending is a wildly used technique in data augment and ensemble models [1]. Unlike MC-dropout, which changes the model slightly and get different predictive results each time, blending transforms input image instance x_0 by rotation, reflection, shift and so on, and makes difference predictions using these transformations. We propose to use the variance of blending predictions for each input sample (named as *blend-Var*) to represent the uncertainty of x_0. $\kappa_f(x_0^i) = var(f(x_0^i))$, while $i = 1, 2, \ldots, T$ and x_0^i denote the ith transformation of x_0.

Blend-Var can be used in NNs with dropout layers or not. To the best of our knowledge, we are the first one to put forward this uncertainty estimation function.

V. Experiments

We evaluate the proposed selective regression method and uncertainty functions on two regression tasks, tropical cyclone (TC) intensity estimation from satellite remote sensing images and apparent age estimation from facial images. We first introduce the network architecture, datasets, and other experimental settings for each regression task, and then present the evaluation results of the proposed uncertainty functions and selective regression models.

A. Two Case Studies

1) Tropical Cyclone (TC) Intensity Estimation: Tropical cyclone (TC) intensity estimation from satellite imagery is a typical regression problem. In this paper, we chose the current state-of-the-art regression model presented in [1], which is a CNN model based on AlexNet without dropout layers as shown in Figure 1. Given an input image, intensity is predicted by following a rotation-blending step, where the image is rotated by different angels as inputs to the model and resultant predictions are averaged as the final predicted intensity. We implemented this rotation-blended model in TensorFlow.

We used the the open benchmark dataset released by [1], Tropical Cyclone for Image-to-intensity Regression (TCIR) dataset, which could be got from https://www.csie.ntu.edu.tw/htlin/program/TCIR, collected from satellite remote sensing. We used 39811 satellite images of TCs in $2003 \sim 2014$ as the training set for training the regression CNN model. We randomly partitioned 11060 satellite images of TCs in $2015 \sim 2017$, and used one half as the validation set for constructing the selective regression model, and the other half as the test set.

For the tropical cyclone (TC) intensity estimation, we used MSE as the loss function, which means $\boldsymbol{Y}_i = (f(x_i) - y_i)^2$, where $f(x_i)$ is the predict value, y_i is the ground truth for the ith instance.

2) Apparent Age Estimation: Apparent Age Estimation, which tries to estimate the age as perceived by other humans from a facial image, is different from the biological (real) age prediction. [24] and [25] built convolutional neural networks (CNNs) based on VGG-16 and achieved the state-of-the-art results for both real and apparent age estimation on the largest apparent age annotation dataset, ChaLearn Looking At People (LAP) dataset [4]. We downloaded their apparent age estimation model trained already on the LAP dataset from https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/. In their model, apparent age estimation was treated as a multi-class classification of age bins, *i.e.,* 101 age bins from 0 to 100, followed by a softmax layer to output final estimated age, as shown in Figure 2. The LAP dataset also provided a validation set of 1043 images and a test set of 1003 images.

Mean absolute error (MAE) in years was used as the quantitative evaluation, which is a default standard evaluation. In this case, let $\boldsymbol{Y}_i = |(f(x_i) - y_i)|$.

B. Choice of κ_f

1) Blend-Var: For the rotation-blended CNN model shown in Figure 1, we cannot use MC-dropout as the uncertainty function, as it does not have any dropout layers. We consider

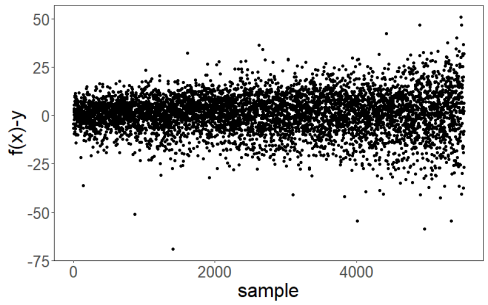


Fig. 3. Regression Errors in Order of κ_f for TC Intensity Estimation.

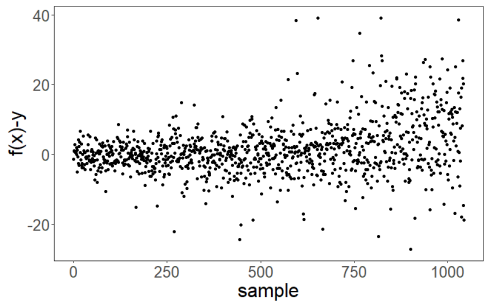


Fig. 4. Regression Errors in Order of κ_f for Apparent Age Estimation.

Blend-Var in this case as our choice of the uncertainty function. Suppose we blend predictions from $T = 10$ rotations for each input image, which means we rotate an image by 0, 36, 72, ... , 288, 324 degrees, then treat them as inputs to the model to obtain $T = 10$ predictions, and calculate the variance as Blend-Var for this image.

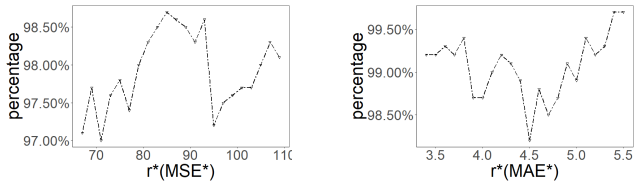
We plot the regression error for each sample in the validation set in order of κ_f with $T = 10$ in Figure 3. We can see that as Blend-Var increases, the average regression errors indeed tend to increase as well, which means our choice of κ_f here is a good estimation of the model uncertainty.

2) MC-dropout: For the apparent age estimation model shown in Figure 2, there are dropout layers in this architecture and no rotation or flip augment method applied, so we consider *MC-dropout* as the uncertainty function and performed $T = 20$ stochastic forward passes through the network as suggested in [25].

We plot the regression error for each sample in the validation set in order of κ_f in Figure 4. We can see that as the MC-dropout variance increases, the average regression errors indeed tend to increase as well, which means our choice of κ_f here is a good estimation of the model uncertainty.

C. Confidence Level

Since we used an approximated $b - a$ in our SGRL algorithm, in this set of experiments we verify whether the empirical risk values are indeed less than the desired risk values r^* on the test set with the given confidence level. For both TCIR and LAP, we split the dataset into two random halves, one for validation and one for testing, for 1,000 times. We then applied the SGRL algorithm on each validation set



(a) TCIR (b) LAP

Fig. 5. Percentage of Empirical Risk $\leq$ Risk Bound on the Test Set over 1,000 Runs

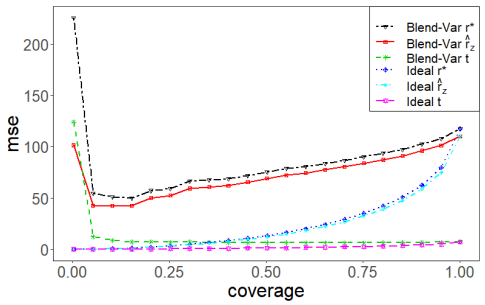


Fig. 6. Risk-Coverage Curve on the Validation Set for TC Intensity Estimation.

with a range of desired risk r^* values and a confidence level $\delta = 0.05$. For each r^* value, we constructed a selective regression model g_θ on each validation set, applied it to the corresponding test set, and calculated the percentage that the empirical risk value on the test set is indeed less than the given risk bound over the 1,000 runs, which is shown in Figure 5. We can see that although we used an approximated $b - a$, the actual percentages are greater than 95% for all desired risk values for both TCIR and LAP, which suggests that the Hoeffding bound with a confidence level can provide a good guidance for constructing selective regression models.

D. Selection Results

In this set of experiments, we examined the selection performance of our SGRL algorithm in more details. For TCIR, we chose one random split of the validation and test sets and evaluated the risk-coverage curve on the validation set and the selection performance on the test set. For LAP, we used the original split of the validation and test sets from the LAP website. Our SGRL algorithm employed both the proposed uncertainty functions and an ideal uncertainty function $\kappa_f = |f(x) - y|$, which ranks each sample in the order of its regression error.

1) Tropical Cyclone Intensity Estimation: We first show the risk-coverage curve obtained by the selective regression model on the validation set for both Blend-Var and the ideal uncertainty function in Figure 6. For each uncertainty function, we sorted samples in the validation set w.r.t. the function and drew three lines of the risk bound r^*, the empirical risk $\hat{r}_z$, and the gap t between r^* and $\hat{r}_z$ in terms of Mean Squared Error (MSE) with increasing coverage. Here we started the curves from the coverage value of 0.01.

TABLE I
MSE RESULTS FOR TCIR WITH $\delta = 0.05$.

RMSE*	r*	val-MSE	val-RMSE	val-Coverage	test-MSE	test-RMSE	test-Coverage
8.5	72.25	65.81	8.11	45.82	64.94	8.06	46.84
9	81	74.47	8.63	60.51	76.47	8.74	61.03
9.5	90.25	83.82	9.16	75.37	85.28	9.23	75.90
10	100	93.33	9.66	88.57	97.08	9.85	89.42
10.25	105	98.18	9.91	91.74	99.87	9.99	92.48
10.44	109	102.02	10.10	95.57	103.19	10.16	96.24
-	-	110.16	10.50	100	109.95	10.49	100

TABLE II
MAE RESULTS FOR LAP WITH $\delta = 0.05$.

MAE* (r*)	val-MAE	val-Cov	test-MAE	test-Cov
4	3.19	42.95	3.46	43.27
4.5	3.88	70.47	4.03	66.54
5	4.40	82.65	4.36	79.48
5.2	4.64	86.67	4.53	86.54
5.5	4.96	93.10	4.86	93.22
-	5.37	100	5.22	100

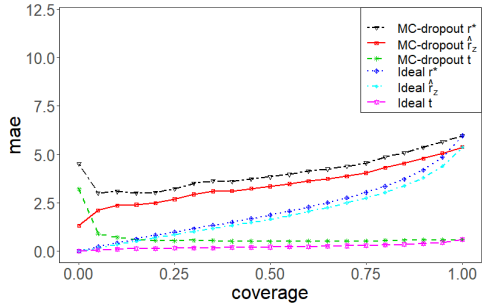


Fig. 7. Risk-Coverage Curve on the Validation Set for Apparent Age Estimation.

There are several observations we can make from for Figure 6. Firstly, for both Blend-Var and the ideal uncertainty function, we can see that the empirical risk bounded by r^* increases when the coverage increases as we expected. Secondly, the gap t between r^* and $\hat{r}_z$ is large with small coverage values as small n made $t = \sqrt{-\frac{(b-a)^2 \ln \delta}{2n}}$ large. When the coverage increases, the gap t becomes stable and relatively small w.r.t. both r^* and $\hat{r}_z$. Therefore, we can use our SGRL algorithm to search for g_θ only when g_θ selects more than m_0 (the minimum selected sample requirement) samples in the validation set. Finally, compared with the perfect risk-coverage curves achieved by the ideal uncertainty function, Blend-Var achieved higher MSE values with the same coverage on the validation set. However, as suggested by Figure 6, Blend-Var can still be used to construct selective regression models that decrease MSE significantly while covering most of the samples.

Given a risk bound r^*, we used the SGRL algorithm to search for g_θ on the validation set, applied the found model to the test set, and calculated the MSEs, Root Mean Square Errors (RMSEs), and coverage values on both validation and test sets for TCIR, which are shown in Table 1. The original model f achieved a RMSE value of 10.50 and 10.49 on the validation and test sets, respectively. As shown in Table 1, the MSE and coverage values are very similar on the validation and test sets. Both val-MSE and test-MSE are bounded by r^* with a gap introduced by the Hoeffding Inequality. Finally, our SGRL algorithm with Blend-Var as the uncertain function can achieve a RMSE value of 9.5 while covering more than 75% samples with a guided confidence level of 95%.

2) Apparent Age Estimation: The risk-coverage curves of the risk bound r^*, the empirical risk $\hat{r}_z$, and the gap t between r^* and $\hat{r}_z$ in terms of Mean Absolute Error (MAE) with increasing coverage for LAP are shown in 7. Trends are similar for this case as well. Because the size of the validation set is smaller (around 1,000) for LAP, the gap between r^* and $\hat{r}_z$ of MC-dropout is larger, which means the risk bound is looser. Nevertheless, MC-dropout can be used to construct selective regression models that decrease MAE significantly while covering most of the samples.

Similarly, we applied the found selective regression model to the test set, and calculated the MAEs and coverage values on both validation and test sets, which are shown in Table II. The original model f achieved a MAE value of 5.37 and 5.22 on the LAP validation set (1043 samples) and the LAP test set (1003 samples), respectively. Again, because the size of the validation set is smaller (around 1,000), the gap between $r*$ and val-MAE, test-MAE is larger. Finally, our SGRL algorithm with MC-dropout as the uncertain function can achieve a MAE value of 4.5 while covering more than 66% samples with a guided confidence level of 95%.

VI. CONCLUSION

We present a general method to construct a selective regression model in deep neural networks (DNNs) that can find out the maximum coverage under the control of the risk. We purpose an uncertainty estimation function, Blend-Var, which could be used in both classification and regression DNNs with blending. We evaluated our proposed method with two real-world applications and achieved promising results. For example, for TC intensity estimation, the proposed selection regression model can achieve a RMSE value of 9.5 for 75% test coverage with a guided confidence level of 0.05, whereas its state-of-the-art model without selection achieved a RMSE value of 10.5.

REFERENCES

[1] B. Chen and B. F. Chen and H. T. Lin, *Rotation-blended CNNs on a new open dataset for tropical cyclone image-to-intensity regression*, in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018.

[2] J. Miller, and M. Maskey and T. Berendes, *Using Deep Learning for Tropical Cyclone Intensity Estimation*, in AGU Fall Meeting Abstracts, 2017.

[3] J. S. Combinido and J. R.Mendoza and J. Aborot, *A Convolutional Neural Network Approach for Estimating Tropical Cyclone Intensity Using Satellite-based Infrared Images*, in 2018 24th International Conference on Pattern Recognition (ICPR), 2018.

[4] S. Escalera and J. Fabian and P. Pardo and X. Baro and J. Gonzalez and H. J. Escalante and D. Misevic and U. Steiner and I. Guyon, *Chalearn looking at people 2015: Apparent age and cultural event recognition datasets and results*, in Proceedings of the IEEE International Conference on Computer Vision Workshops, 2015.

[5] G. Levi and T. Hassner, *Age and gender classification using convolutional neural networks*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2015.

[6] A. S. Qureshi and A. Khan and A. Zameer and A. Usman, *Wind power prediction using deep neural network based meta regression and transfer learning*, Applied Soft Computing, 58 (2017), pp. 742–755.

[7] J. Zhou and X. Hong and F. Su and G. Zhao, *Recurrent convolutional neural network regression for continuous pain intensity estimation in video*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2016.

[8] C. K. Chow, *An optimum character recognition system using decision functions*, IRE Transactions on Electronic Computers, (1957), pp. 247–254.

[9] L. P. Cordella and C. De Stefano and F. Tortorella and M. Vento, *A method for improving classification reliability of multilayer perceptrons*, IEEE Transactions on Neural Networks, 6 (1995), pp. 1140–1147.

[10] C. De Stefano, and C. Sansone and M. Vento, *To reject or not to reject: that is the question-an answer in case of neural classifiers*, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 30 (2000), pp. 84–94.

[11] R. El-Yaniv and Y. Wiener, *On the foundations of noise-free selective classification*, Journal of Machine Learning Research, 11 (2010), pp. 1605–1641.

[12] Y. Geifman and R. El-Yaniv, *Selective classification for deep neural networks*, in Advances in Neural Information Processing Systems, 2017.

[13] R. Pradhan and R. S. Aygun and M. Maskey and R. Ramachandran and D. J. Cecil, *Tropical cyclone intensity estimation using a deep convolutional neural network*, IEEE Transactions on Image Processing, 27 (2017), pp. 692–702.

[14] C. Chow, *On optimum recognition error and reject tradeoff*, IEEE Transactions on information theory, 16 (1970), pp. 41–46.

[15] Y. Geifman and R. El-Yaniv, *SelectiveNet: A Deep Neural Network with an Integrated Reject Option*, arXiv preprint arXiv:1901.09192, 2019.

[16] Y. Gal and Z. Ghahramani, *Dropout as a bayesian approximation: Representing model uncertainty in deep learning*, in International Conference on Machine Learning, 2016.

[17] N. Srivastava, and G. Hinton and A. Krizhevsky and I. Sutskever and R. Salakhutdinov, *Dropout: a simple way to prevent neural networks from overfitting*, The journal of machine learning research, 15 (2014), pp. 1929–1958.

[18] G. S. Babu and P. Zhao and X. L. Li, *Deep convolutional neural network based regression approach for estimation of remaining useful life*, in International Conference on Database Systems for Advanced Applications, 2016.

[19] R. Herbei and M. H. Wegkamp, *Classification with reject option*, Canadian Journal of Statistics, 34 (2006), pp. 709–721.

[20] W. Hoeffding, *Probability inequalities for sums of bounded random variables*, in The Collected Works of Wassily Hoeffding, 1994.

[21] N. Cesa-Bianchi and G. Lugosi, *Prediction, learning, and games*, Cambridge university press, 2006.

[22] L. Zhao and L. Wang and D. Cui, *Hoeffding bound based evolutionary algorithm for symbolic regression*, Engineering Applications of Artificial Intelligence, 25 (2012), pp. 945–957.

[23] O. Maron, *Hoeffding Races–model selection for MRI classification*, Massachusetts Institute of Technology, 1994.

[24] R. Rothe, R. Timofte, L. V. Gool, *DEX: Deep EXpectation of apparent age from a single image*, in IEEE International Conference on Computer Vision Workshops (ICCVW), 2015.

[25] R. Rothe, R. Timofte, L. V. Gool, *Deep expectation of real and apparent age from a single image without facial landmarks*, International Journal of Computer Vision (IJCV), 126 (2016), pp. 144–157.

[26] R. Gelbhart and R. El-Yaniv, *The Relationship Between Agnostic Selective Classification, Active Learning and the Disagreement Coefficient.*, Journal of Machine Learning Research, 20 (2019), pp. 1–38.